Operator RECU_FONCTION

1 Goal

To extract in the form of a function the evolution from a size according to another.

If the retrieval is carried out starting from a structure of data `result`, or of a field of size `cham_gd`, or of one `resu_gene`, the produced function corresponds to the temporal evolution of a component in a node or a point of Gauss of the grid.

Of a structure of data `tran_gene`, one can also extract the evolution from two parameters in a node of shock.

Of one `table`, one can extract the evolution from 2 parameters in the columns of the table or a function contained in a box of the table.

Of one `tablecloth`, one can extract the function corresponding to a value given from the parameter.

Of a structure of data `melasflu` one can extract the evolution from modal parameters according to the rate of flow of the fluid or evolution critical velocity according to the number of Connors.

Of a structure of data `interspectre` one can extract, the frequential evolution of the interspectre associated with i - ième line j - ième column of the matrix interspectrale or the temporal evolution of the interspectre with n - ième component of the diagonal.

Product a structure of data of the type `function` or `fonction_c`.

At exit of the order, the function is reordered by increasing X-coordinates. On the other hand, it is interdict to have several identical X-coordinates (it would not be any more one function).

2 Syntax

```
FR [function]=  RECU_FONCTION    (
    ♦  /  RESULT = resu,                                  /  [dyna_harmo]
                                                          /  [evol_elas]
                                                          /  [dyna_trans]
                                                          /  [evol_ther]
                                                          /  [evol_noli]
          # See extraction and localization of the field
       /  CHAM_GD = ch_gd,                                /  [cham_no]
                                                           /  [cham_elem]
          # See operands of localization of the field
       /  RESU_GENE = embarrassment,                         /  [tran_gene]
          # Temporal evolution of a physical component
             ♦  /  NOM_CHAM      =/'DEPL',
                                   /'QUICKLY',
                                   /'ACCE',
                                   /'PTEM',
                   ♦  NOM_CMP  = cmp,                      [K]
                   /  GROUP_NO = grno,                     [gr_noeud]
                   /  NUME_CMP_GENE = val_n                [I]
                   ◊  /     MULT_APPUI = 'YES',
                      /  |  CORR_STAT =  'YES',
                         |  ACCE_MONO_APPUI =   frap,      [function]
                 /  GROUP_NO_CHOC= no_choc,                [gr_noeud]
                   ♦  PARA_X       = nparax,               [KN]
                   ♦  PARA_Y       = nparay,               [KN]
                   ◊  ENTITLE      = name,                    [KN]
                   ◊  LIST_PARA    = will li_para,            [listr8]
                   ◊  SOUS_STRUC   = nom_str,              [KN]
    /  RESU_GENE = embarrassment,                             [harm_gene]
    # Frequential evolution of a generalized or physical component
          ♦  NOM_CHAM      = nomsymb,                     [K16]
          ♦  /  NUME_CMP_GENE = numcmp,                   [K8]
             /  NOM_CMP = cmp,                            [K]
                ♦  GROUP_NO = grno,                    [gr_noeud]
    /  RESU_GENE = embarrassment,                             [mode_gene]
    # Frequential evolution of a generalized or physical component
          ♦  /  NOM_PARA_RESU = parameter,                [K8]
             /  NOM_CHAM      = nomsymb,                  [K16]
                ♦  /   NUME_CMP_GENE = numcmp,            [K8]
                   /   NOM_CMP = cmp,                     [K]
                      ♦  GROUP_NO = grno,              [gr_noeud]
                      ♦  /  SKELETON  = squ,           [skeleton]
                         /  SOUS_STRUC   = sstru,      [K]
```

```
~   TABLE =   tabl,                                  /   [table]
          ♦   PARA_X    =  nparax,                       [KN]
          ♦   PARA_Y    =  nparay,                       [KN]
          ◊   NOM_PARA_TABL    =  / 'FUNCTION',          [KN]
                                  / 'FONCTION_C'
              ◊   FILTER =  _F (
                        ♦   NOM_PARA  =                  [KN]
                        ◊   CRIT_COMP=   /  'EQ',        [DEFECT]
                                         /  'LT',
                                         /  'WP',
                                         /  '',
                                         /  'IT',
                                         /  'GE',
                                        /  'VACUUM',
                                        /  'NON_VIDE',
                                        /  'MAXIMUM',
                                        /  'MAXI_ABS',
                                        /  'MINI',
                                        /  'MINI_ABS',
                        ♦   /  VALE    = val_r,          [R]
                            /  VALE_I  = val_n,          [I]
                            /  VALE_C  = val_c,          [C]
                            /  VALE_K  = val_k,          [KN]
                        ◊   |  CRITERION =  /  'RELATIVE',  [DEFECT]
                                            /  'ABSOLUTE',
                            |  PRECISION =  /  prec,
                                            /  0,001,    [DEFECT]
                               ),
    /   BASE_ELAS_FLUI = flui,                           [melasflu]
        ♦   NUME_MODE   =    im,                         [I]
        ♦   /   PARA_X      =    /'VITE_FLU',           [KN]
                ♦   /  TOUT_ORDRE   =  'YES',           [DEFECT]
                /   NUME_ORDRE   =  is,                 [I]
                ♦   PARA_Y       =  /'FREQ',            [KN]
                                    /'AMOR',
            /   PARA_X      =    /'NB_CONNORS',         [KN]
                ♦   PARA_Y       = 'VITE_CRIT',         [KN]
    /   INTE_SPEC = intespec,                            [interspectre]
  # Frequential evolution of a component of the matrix interspectrale
        ◊   NOM_CHAM      = nomsymb,                     [K16]
        ♦   /   ♦   NUME_ORDRE_I   = numei,              [I]
                ◊   NUME_ORDRE_J   = numej,              [I]
            /   ♦   NUME_ORDRE   =  numei,               [I]
            /   ♦   NOEUD_I    = noei,                   [node]
                ♦   NOM_CMP_I  = cmpi,                   [KN]
                ◊   NOEUD_J    = noej,                   [node]
                ◊   NOM_CMP_J  = cmpj,                   [KN]
    /   TABLECLOTH = nap,                                      [tablecloth]
        ♦   VALE_PARA_FONC = Np,                         [KN]
        ◊   |  CRITERION =  /  'RELATIVE',                  [DEFECT]
                            /  'ABSOLUTE',
            |  PRECISION =  /  prec,
                            /  0,001,                    [DEFECT]

  # Operands of extraction of the field or the parameter if one
handle one SD_resultat or RESU_GENE or one cham_gd

        ♦   /   NOM_CHAM      = nomsymb,                 [K16]
```

```
                /  NOM_PARA_RESU = parameter,
            ◊   /  TOUT_ORDRE   =  'YES',              [DEFECT]
                /  TOUT_INST    =  'YES',
                /  NUME_ORDRE   =  l_nume,             [l_I]
                /  LIST_ORDRE   =  l_ord,              [listis]
                /  /  INST      =  l_inst,             [l_R]
                   /  LIST_INST =  li_inst,            [listr8]
                   /  FREQ      =  l_freq ,            [l_R]
                   /  LIST_FREQ =  li_freq,            [listr8]
                   ◊  |  PRECISION =  /  prec,         [R]
                                      /  1.0D-3,       [DEFECT]
                      |  CRITERION   =  /  'RELATIVE',  [DEFECT]
                                      /  'ABSOLUTE',
                   ◊  INTERP_NUME  =  /  'NOT',        [DEFECT]
                                      /  'FLAX',
```

Operands of localization of the field if one is handled `result` or one `cham_gd`

```
            ♦   /  GROUP_NO = grno,                [gr_noeud]
                /  GROUP_MA  = grma,               [gr_maille]
                      ◊  NOT    = nupoint,      [I]
                      ◊  SOUS_POINT   = nusp,      [I]
            ♦   NOM_CMP = cmp,                     [K]
```

Overload of the attributes of the function created

```
            ◊   NOM_PARA = nom_pa,                 [KN]
            ◊   NOM_RESU = nom_res,                [KN]
            ◊   Interpol =   /  'NOT',             [KN]
                             /  |  'FLAX',
                                |  'LOG',
            ◊   PROL_DROITE =   /  'CONSTANT',
                                /  'LINEAR',
                                /  'EXCLUDED',
            ◊   PROL_GAUCHE =   /  'CONSTANT',
                                /  'LINEAR',
                                /  'EXCLUDED',
      ◊  TITLE =   T,                                  [l_K]
      ◊  INFORMATION =   /  1,                               [DEFECT]
                         /  2,
                                    )
```

If `RESULT` is one [`dyna_harmo`] then `Fr` is [`fonction_c`],
If `RESU_GENE` is one [`harm_gene`] then `Fr` is [`fonction_c`],
If `INTE_SPEC` and `NUME_ORDRE_J` or `NOEU_J` then `Fr` is [`fonction_c`],
If `NOM_PARA_TABL` = `'FONCTION_C'` then `Fr` is [`fonction_c`],
In the other cases, `Fr` is [`function`].

3 Operands

3.1 Operand RESULT

♦ RESULT = resu

Name of the concept of the type result to which the extraction relates.

For the operands allowing to extract the field, to refer to [§3.7].

For the operands allowing to locate the field, to refer to [§3.8].

3.2 Operand CHAM_GD

♦ CHAM_GD = ch_gd

Name of the concept of a field to which the extraction relates. For the operands allowing to locate the field, to refer to [§3.7].

The field provided to the keyword CHAM_GD is:
- that is to say a field with the nodes of size: DEPL_R, TEMP_R or PRES_R ;
- that is to say a field by elements (with the nodes or the points of Gauss) of size: VARI_R, EPSI_R, FLUX_R, or PRES_R.

3.3 Operand RESU_GENE

3.3.1 Temporal evolution of a physical or generalized, standard component tran_gene

♦ RESU_GENE = embarrassment

Name of the concept of the type resu_gene product by DYNA_TRAN_MODAL [U4.53.21] to which the extraction relates.

The recovered function is expressed
- according to the physical variables: one specified GROUP_NO = grno.
- according to the generalized variables: one specified NUME_CMP_GENE = n_vaL.

NOM_CHAM give the name of the field which one wants to recover ('DEPL', 'QUICKLY', 'ACCE', or 'PTEM'). The option 'PTEM' allows to extract, for each moment (or sequence number) of recovery, the step values of computing time.

Note:

> *To make this restitution on physical basis is the role of the order REST_GENE_PHYS. In the later versions, this functionality will be withdrawn from RECU_FONCTION, it will be necessary to make the restitution then extract the function.*

3.3.1.1 Operands MULT_APPUI and ACCE_MONO_APPUI

◊ MULT_APPUI

If this keyword is 'YES', one in the case of restores the evolution of the variables in physical space by dealing with the problem moving absolute an excitation multi-support. In the contrary case, the restitution in physical space is done by supposing that with the problem is dealt moving relative. This keyword is not usable if the keyword CORR_STAT is used.

◊ ACCE_MONO_APPUI

In the case of an acceleration mono-support, one must indicate here the acceleration imposed on all the supports in the direction considered in order to calculate the absolute acceleration of the point.

If the keyword is not indicated, one obtains relative acceleration as a result of the order.

Note:

The name of the concept must be same as that well informed under FONC_MULT of DYNA_TRAN_MODAL.

3.3.1.2 Operands CORR_STAT

◊ CORR_STAT

If this keyword is 'YES', the evolution of the variables in physical space is obtained by taking account of the correction due to the catch in consideration of static modes (cf [R4.05.03]). This keyword is not usable if the keyword MULT_APPUI is used.

3.3.1.3 Information concerning the nodes of shock

♦ RESU_GENE = embarrassment

Concept of the type tran_gene containing for the various nodes of shock: local displacements, normal and tangential speeds and normal and tangential forces of shock.

♦ GROUP_NO_CHOC = no_choc,

Groupe of nodes (which contains one node) shock where the function is recovered.

This node of shock is defined in the order DYNA_TRAN_MODAL [U4.53.21].

♦ PARA_X = nparax

Name of the parameter defining the X-coordinates (argument taken among the list: 'INST', 'FN', 'FT1, 'FT2', 'VN', 'VT1', 'VT2', 'DXLOC', 'DYLOC', 'DZLOC', 'VINT1', 'VINT2', 'VINT3', 'VINT4', 'VINT5', 'VINT6', 'VINT7', 'VINT8', 'VINT9', 'VINT10', 'VINT11', 'VINT12', 'VINT13', 'VINT14', 'VINT15', 'VINT16', 'VINT17', 'VINT18', 'VINT19', 'VINT20', 'VINT21').

♦ PARA_Y = nparay

Name of the parameter defining the ordinates (argument taken among the list: 'INST', 'FN', 'FT1, 'FT2', 'VN', 'VT1', 'VT2', 'DXLOC', 'DYLOC', 'DZLOC', 'VINT1', 'VINT2', 'VINT3', 'VINT4', 'VINT5', 'VINT6', 'VINT7', 'VINT8', 'VINT9', 'VINT10', 'VINT11', 'VINT12', 'VINT13', 'VINT14', 'VINT15', 'VINT16', 'VINT17', 'VINT18', 'VINT19', 'VINT20', 'VINT21').

◊ LIST_PARA = will li_para

List of the values of the parameter in X-coordinate defining the function.
Caution: It is possible to have problems of interpolation of the result on this parameter list (because except for the precision machine, the values can be slightly out of terminals of the produced functions). In this case, it is enough not to use this keyword here. The function will then be created over every moment of calculation. One can then interpolate this function with CALC_FONC_INTERP on the list of the parameters of its choice by controlling the prolongations on the left and on the right.

◊ ENTITLE = name

This name defines the connection of shock (this name if it is used, is defined in the order DYNA_TRAN_MODAL [U4.53.21]).

◊ SOUS_STRUC = nom_str

During a calculation in dynamic under-structuring, name of the substructure which contains the node of shock (cf orders DEFI_MODELE_GENE [U4.65.02]). In this case the keyword ENTITLE must be also well informed.

3.3.2 Frequential evolution of a generalized or physical, standard component harm_gene

♦ RESU_GENE = embarrassment

Name of the concept of the harm_gene type produces by DYNA_LINE_HARM [U4.53.11].

The recovered function is expressed
- according to the physical variables: one specified GROUP_NO = grno.
- according to the generalized variables: one specified NUME_CMP_GENE = n_vaL.

NOM_CHAM gives the name of the field which one wants to recover ('DEPL', 'QUICKLY' or 'ACCE').

Note:

> *To make this restitution on physical basis is the role of the order REST_GENE_PHYS . In the later versions, this functionality will be withdrawn from RECU_FONCTION , it will be necessary to make the restitution then extract the function.*

3.3.3 Frequential evolution of a generalized or physical, standard component mode_gene

♦ RESU_GENE = embarrassment

Name of the concept of the type mode_gene product by CALC_MODES [U4.53.02].
The recovered function is expressed with the physical variables if NOM_CMP is present, with the generalized variables if NUME_CMP_GENE is present.

♦ NOM_PARA_RESU/NOM_CHAMP See paragraph 3.8.
♦ NOM_CMP/GROUP_NO See paragraph 3.9.

♦ SKELETON Name of the grid skeleton of the total structure on which the result will be restored: to see the operator DEFI_SQUELETTE [U4.24.01].
♦ SOUS_STRUC See above.

3.4 Operand TABLE

One can recover:
1) that is to say a function defined starting from two columns of the table,
2) that is to say a function whose name is indicated in a box of the table.

♦ TABLE = tabl Name of the table result in which one carries out an extraction.

3.4.1 Function defined starting from two columns of the table

3.4.1.1 Operands PARA_X / PARA_Y

♦ PARA_X = nparax

Name of the column of the table defining the X-coordinates.

♦ PARA_Y = nparay

Name of the column of the table defining the ordinates.

3.4.2 Function whose name is indicated in a box of the table

3.4.2.1 Operand NOM_PARA_TABL

♦ NOM_PARA_TABL= 'FUNCTION' or 'FONCTION_C'

The presence of this keyword indicates that one recovers a function whose name is stored in a box of the table. The real functions are stored in the column 'FUNCTION', complex functions in the column 'FONCTION_C'.

3.4.2.2 Keyword FILTER

◊ FILTER

The operands of extraction are different from those used for the preceding cases. To carry out the extraction, the keyword should be used FILTER and operands NOM_PARA, CRIT_COMP, VALE_X, CRITERION, PRECISION.

This keyword factor makes it possible to filter the information stored in the table. For the use of this keyword to see the order IMPR_TABLE [U4.91.03].

To recover a function whose name is indicated in a box of the table, the keyword factor should be used at least twice FILTER to select only the useful box.

3.5 Operand BASE_ELAS_FLUI

One recovers in a structure of data of the type melasflu produced by the operator CALC_FLUI_STRU [U4.66.02], evolutions of the frequency or damping, for a given mode, according to various speeds of excitation of the fluid.

♦ BASE_ELAS_FLUI = flui

Concept of the type melasflu product by the order CALC_FLUI_STRU.

3.5.1 Operand NUME_MODE

♦ NUME_MODE = im

Number of the mode for which the extraction of the frequency or damping according to the speed of the fluid is carried out.

3.5.2 Operands PARA_X / PARA_Y/NUME_ORDRE/TOUT_ORDRE

♦ PARA_X =/'VITE_FLU'
/'NB_CONNORS'

In X-coordinate, the parameter is the speed of excitation of the fluid, of name 'VITE_FLU' or well the number of Connors, name 'NB_CONNORS'.

If one chose like X-coordinate the speed of excitation of the fluid 'VITE_FLU' , then:

♦ PARA_Y = / 'FREQ',
/ 'AMOR'.

In ordinate, one has the choice between the frequency (name of the parameter 'FREQ') or damping (name of the parameter 'AMOR').

♦ / NUME_ORDRE = is,
/ TOUT_ORDRE = 'YES',

The evolution of the frequency or that of damping is given for all speeds of the fluid (TOUT_ORDRE) or for some sequence numbers speeds of the fluid (NUME_ORDRE).

If one chose like X-coordinate the number of Connors 'NB_CONNORS', then :

♦ PARA_Y = / 'VITE_CRIT'.

In ordinate, one obtienT critical velocity [R4.07.04].

3.6 Operand INTE_SPEC

One extracts in a structure of data of the type interspectre, frequential evolution of the interspectre associated with line i and column j matrix interspectrale intespec. One can also extract the temporal evolution from one of the functions generated by GENE_FONC_ALEA [U4.36.05] by specifying its sequence number.

♦ INTE_SPEC = intespec

3.6.1 Operand NOM_CHAM

◊ NOM_CHAM = nomsymb

Reference symbol of the field to which the extraction relates.

3.6.2 Operands NUME_ORDRE_I, NUME_ORDRE_J

♦ NUME_ORDRE_I = numei
◊ NUME_ORDRE_J = numej

Indication of the couple of indices (line i, column j) to extract from the matrix interspectrale intespec.

These operands are excluded with the operands NUME_ORDRE NOEUD_I NOM_CMP_I NOEUD_J NOM_CMP_J.

Note:

It is not obligatory to inform NUME_ORDRE_J if one wants to extract a term from the diagonal of the matrix.

3.6.3 Operands NUME_ORDRE

♦ NUME_ORDRE = digital

Indication of the sequence number of the temporal function generated by the operator GENE_FONC_ALEA [U4.36.05].

These operands are excluded with the operands NUME_ORDRE_I NUME_ORDRE_J NOEUD_I NOM_CMP_I NOEUD_J NOM_CMP_J.

3.6.1 Operands NOEUD_I, NOM_CMP_I, NOEUD_J, NOM_CMP_J

♦ NOEUD_I = noei
♦ NOM_CMP_I = cmpi
◊ NOEUD_J = noej
◊ NOM_CMP_J = cmpj

These operands correspond to the names of the nodes and the components (line I, column J) of the matrix of the matrix interspectrale intespec.

These operands are excluded with the operands NUME_ORDRE_I NUME_ORDRE_J NUME_ORDRE.

Note:

It is not obligatory to inform NOEUD_J and NOM_CMP_J if one wants to extract a term from the diagonal of the matrix.

3.7 Operand TABLECLOTH

One recovers in a structure of data of the type tablecloth the function corresponding to a given value of the parameter of the tablecloth.

♦ VALE_PARA_FONC = Np

np is the value of the parameter of the tablecloth for which one wishes to extract the function.
There is no interpolation on the parameter of the tablecloth. CRITERION and PRECISION allow to provide np with a given precision.

3.8 Operands of extraction of the field or the parameter

3.8.1 Operand NOM_CHAM

♦ NOM_CHAM = nomsymb

Reference symbol of the field to which the extraction relates.

3.8.2 Operand NOM_PARA_RESU

♦ NOM_PARA_RESU = parameter

Reference symbol of the structural parameter of data which one wants to extract (for example: ETA_PILOTAGE, MASSE_EFFE_DX, MASSE_GENE …).

The extracted function will then have as a X-coordinate the variable of access (INST, FREQ…) and for ordinate the value of parameter.

3.8.3 Operands TOUT_ORDRE/NUME_ORDRE/TOUT_INST/LIST_ORDRE

♦ / TOUT_ORDRE = 'YES' (value by default)

This keyword indicates that one wants to extract for all the already calculated sequence numbers.

Example: every moment for a result of the type evol_*.

/ NUME_ORDRE = l_nume

The extraction will be done for the values of sequence number l_nume provided.

/ TOUT_INST = 'YES'

This keyword indicates that one wants to extract for every moment.

/ LIST_ORDRE = l_ord

This keyword indicates that one wants to extract with the sequence numbers described in the concept l_ord of type listis.

3.8.4 Operands INST/LIST_INST/FREQ/LIST_FREQ

◊ / INST = l_inst

This keyword indicates that one wants to extract at the moments l_inst.

/ LIST_INST = li_inst

This keyword indicates that one wants to extract at the moments described in the concept li_inst of type listr8.

/ FREQ = l_freq

This keyword indicates that one wants to extract at the frequencies l_freq.

/ LIST_FREQ = li_freq

This keyword indicates that one wants to extract at the frequencies described in the concept li_freq of type listr8.

3.8.5 Operands PRECISION/CRITERION

◊ PRECISION = prec

This operand makes it possible to indicate that one searches the value of the field whose moment or frequency is in an interval defined by the absolute or relative position: "inst ± prec"(Cf. CRITERION).

By default prec = 1.0D-3

◊ CRITERION =

'RELATIVE' the interval of research is
[inst (1-prec), inst (1+prec)]

'ABSOLUTE' the interval of research is
[inst-prec, inst+prec]

3.8.6 Operand INTERP_NUME

◊ INTER_NUME
This keyword defines the type of interpolation between two sequence numbers. It is valid only if the user defined a list of moments or frequencies. It is possible to prohibit the interpolation 'NOT' or to admit a linear interpolation 'FLAX'.
The interpolation cannot be used when one extracts the value from a parameter (keyword NOM_PARA_RESU).

3.9 Operands of localization of the field

3.9.1 Operand GROUP_NO

♦ GROUP_NO = grno

Name of the group of nodes, containing 1 only node, to which the extraction relates.

3.9.2 Operands GROUP_MA/GROUP_NO/NOT

♦ GROUP_MA = grma

Name of a group of meshs (grma), containing only one nets, to which the extraction relates. These keywords relate to only them cham_elem.

♦ / GROUP_NO = grno

Indicate the name of the group of nodes, container only one name of node, to which the extraction (case relates of cham_elem with the nodes).

/ NOT = nupoint

Entirety nupoint specify the local number with the element of the point of GAUSS of which one wishes to obtain the value (case of cham_elem at the points of GAUSS).

◊ SOUS_POINT = nusp

Entirety nusp specify the number of the under-point of which one wishes to obtain the value (case of cham_elem under-points, used by the elements of structure: beam, pipes, hulls).

In the case of the plates and of the multi-layer hulls, the number of the under-point corresponds to the level in the whole of the layers. Each layer is described by a lower, average and higher skin. By convention, for N layers, this number varies between 1 and $3N$ where the first point is at the level of the lower skin of the first layer and it $3N$ ième point on the level of the higher skin of the last layer (cf. [R3.07.03] and [R3.07.04] for the numeration of the layers).
In the case of the multifibre beams, this entirety is the number of the fibre whose classification is described in documentation [U4.26.01] and [R3.08.08].
In the case of the pipes, it is necessary to refer to the description made in the document [R3.08.06].

3.9.3 Opérandbe NOM_CMP/NOM_VARI

◊ NOM_CMP = nocmp, [KN]

Name of the component to which the extraction relates .

◊ NOM_ VARI = novari , [KN]

For the fields of the internal variables (VARI_*), one can give the name of the internal variable to which the extraction relates (see [U4.51.11] for the rules of naming of the internal variables).

3.10 Attributes of the concept function created by RECU_FONCTION

3.10.1 Values by default

By default attributes of the concept function created by the order RECU_FONCTION are:

Interpolation: 'NOT'
Left prolongation: 'EXCLUDED'
Right prolongation: 'EXCLUDED'
NOM_PARA : given as starter
NOM_RESU : given as starter

3.10.2 Overload of the attributes

The user can overload the attributes given by default by using the following keywords:

3.10.2.1 Operand NOM_PARA

◊ NOM_PARA = para

It indicates the name of the parameter (variable or X-coordinate) of the function. Values currently authorized for will lpara are:

/ 'TEMP' / 'INST' / 'EPSI'
/ 'X' / 'Y' / 'Z'
/ 'FREQ' / 'SWEATERS' / 'AMOR'
/ 'DX' / 'DY' / 'DZ'
/ 'DRX' / 'DRY MARTINI' / 'DRZ'

more those specific to the nodes of shock (cf [§ 3.3.2.2]).

3.10.2.2 Operand NOM_RESU

◊ NOM_RESU = resu

It indicates the name of the result, the function thus created is a function whose value is of name lresu (8 characters).

3.10.2.3 Operand Interpol

◊ Interpol

Type of interpolation of the function enters the values of the parameter of the field of definition. Behind this keyword one expects a parameter list (two at the most) among 'NOT', 'FLAX', 'LOG'. If only one value is given the interpolation will be identical for the X-coordinates and the ordinates. If two values are given, the first corresponds to the interpolation of the X-coordinates and the second with the interpolation of the ordinates.

3.10.2.4 Operands PROL_DROITE/ PROL_GAUCHE

◊ PROL_DROITE and PROL_GAUCHE

They define the type of prolongation on the right (on the left) of the field of definition of the variable:

- 'CONSTANT' for a prolongation with the last (or first) value of the function,
- 'LINEAR' for a prolongation along the first definite segment (PROL_GAUCHE) or of the last definite segment (PROL_DROITE),
- 'EXCLUDED' if the extrapolation of the values apart from the field of definition of the parameter is prohibited.

3.11 Operand TITLE

◊ TITLE

Title attached to the concept produced by this operator [U4.03.01].

3.12 Operand INFORMATION

◊ INFORMATION

Specify the options of impression on the file MESSAGE.

1 pas d' impression (by default)

2 impression of the descriptor of the function and the list of the first 10 values of the function in the order ascending of the first 10 parameters

4 Examples

4.1 Extractions of function on the dynamic response of a network of piping

```
tran_gen = DYNA_TRAN_MODAL (…)

l_inst = DEFI_LIST_REEL (BEGINNING = 0. ,
                         INTERVAL = _F (JUSQU_A = 3. , NOT = 0,005))

dyn_tran = REST_GENE_PHYS (RESU_GENE = tran_gen, NOM_CHAM = 'DEPL',
                           LIST_INST = l_inst,   INTERPOL = 'FLAX')

dyn_tran = CALC_CHAMP (…,
                       CONSTRAINT = 'SIEF_ELGA'                    )


tab_rele = POST_RELEVE_T (ACTION=_F (ENTITLES   = 'sixx_254',
                                     WAY = line,
                                     RESULT = dyn_tran,
                                     NOM_CHAM   = 'SIEF_ELGA',
                                     INST    = 2.54,
                                     TOUT_CMP   = 'YES',
                                     OPERATION  = 'EXTRACTION'  )  )
```

4.1.1 Evolution of the displacement of the node NO01 component 'DX' at every moment of calculation

```
f1 = RECU_FONCTION (RESU_GENE = tran_gen,   NOM_CHAM = 'DEPL',
                    GROUP_NO     = 'GNO01'  ,   NOM_CMP = 'DX'     )
```

4.1.2 Evolution of the size 'SIXX' on the mesh MA01 with the node NO01 at every moment of calculation

```
f2 = RECU_FONCTION (RESULTAT= dyn_tran, NOM_CHAM= 'SIEF_ELGA',
                    GROUP_MA  = 'GMA01'  , GROUP_NO   = 'NO01',
                     NOM_CMP=' SIXX')
```

4.1.3 Evolution of the size 'SIXX' along the line of piping at the moment of calcuL 2.54 S

```
f3 = RECU_FONCTION (TABLE = tab_rele,
                    PARA_X = 'ABSC_CURV',  PARA_Y = 'SIXX')
```

4.1.4 Evolution of the size 'SIXX' along the line of piping (curvilinear X-coordinate higher than 10) at the moment of calbottom 2.54 S

```
f4 = RECU_FONCTION (TABLE = tab_rele,
                    FILTER = _F (NOM_PARA = 'ABSC_CURV',
                                 CRIT_COMP = 'GE',
                                 VALE      = 10. ,  ),
                    PARA_X = 'ABSC_CURV',  PARA_Y = 'SIXX')
```

4.2 Extraction of function in a structure of data melasflu

```
meles1   = CALC_FLUI_STRU (  …  )

f_freq   = RECU_FONCTION (  BASE_ELAS_FLUI = meles1,
                            PARA_X         = 'VITE_FLU',
                            PARA_Y         = 'FREQ',
                            TOUT_ORDRE     = 'YES',
                            NUME_MODE      = 2
                          )
```

4.3 Extraction of `function` in a structure of data `interspectre`

```
reppx_ac  = REST_SPEC_PHYS (…)

statx_ac  = POST_DYNA_ALEA (INTERSPECTRE = _F (INTE_SPEC  = reppx_ac,
                                               OPTION     = 'DIAG',
                                              )
                           )

f_freq   = RECU_FONCTION (  INTE_SPEC    = statx_ac,
                            NOEUD_I      = 'N_TUB_01',
                            NOM_CMP_I    = 'DX',
                          )
```